IN THE CLAIMS

1. (Currently Amended) A method for data forwarding within a processor architecture of the type having an array of pipelines and a register file, comprising the steps of:

architecting data from write-back stages of the pipelines to a first section of n registers of the register file;

writing speculative data from the pipelines to ~~first~~ a second section of m registers of the register file~~,~~;

reading the speculative data from the ~~first~~ second section of m registers based upon an age of the speculative data~~,~~; and

forwarding the speculative data to the pipelines to bypass data hazards therein.

2. (Original) The method of claim 1, further comprising the step of processing instructions through the pipelines.

3. (Currently Amended) The method of claim 2, further comprising the step of architecting data to ~~second~~ the first section of n registers ~~of the register file~~ after processing one of the instructions through a write-back stage of one of the pipelines.

4. (Currently Amended) The method of claim 1, further comprising utilizing decode register file column logic of the register file to architect speculative data within the ~~first~~ second section of m registers without moving data.

5. (Currently Amended) The method of claim 4, the decode register file column logic decoding and selecting ~~an~~ the age of the speculative data to enable architecting of the speculative data.

6. (Currently Amended) The method of claim ~~5~~ 4, the decode register file column logic determining whether the speculative data has a newest age.

7. (Currently Amended) The method of claim ~~5~~ 4, the decode register file column logic determining whether a particular column is selected for one of read or write operations.

8. (Original) A processor architecture for bypassing data hazards, comprising (a) an array of pipelines, each of the pipelines having an array of execution units, (b) a register file having a first section of n registers and a second section of m registers, and (c) a read mux for coupling speculative data from the execution units to the second set of m registers and for coupling the speculative data from the second set of m registers to the execution units, to bypass data hazards within the execution units.

9. (Original) The processor architecture of claim 8, further comprising a write mux for coupling non-speculative data from a write-back stage of the execution units to the first section of n registers.

10. (Original) The processor architecture of claim 9, further comprising a first bus structure for communicating the non-speculative data between the execution units and the write mux.

11. (Original) The processor architecture of claim 10, the first bus structure communicating the speculative data between the execution units and the write mux.

12. (Original) The processor architecture of claim 8, the register file comprising column decode logic for each of the registers in the second section of m registers, for architecting speculative data within the second section of m registers without moving data.

13. (Original) The processor architecture of claim 12, the column decode logic decoding speculative data to determine an age therewith.

14. (Original) The processor architecture of claim 13, the column decode logic comprising an age decoder for determining the age.

15. (Original) The processor architecture of claim 13, the column decode logic (a) determining whether the age is a newest age and whether a register associated with the decode logic is selected for a write or read operation, and (b)

architecting the register if the data is the newest and the register is selected for the write or read operation.

16. (Original) The processor architecture of claim 15, the column decode logic comprising a write decoder for architecting the register.

17. (Original) The processor architecture of claim 8, the section of n registers comprising 128 registers.

18. (Original) The processor architecture of claim 8, the section of m registers comprising 16 registers.